Enan Jahic

Reflexion plan vs Reality   Server

Since my class diagram of the Server architecture wasn't closely well considered as the Clients one I had to change a lot more things there.

One of the first thing that I had to change was the way I was validating my gamerules. In the class diagram it used to be that the each rule has its own method which returns Boolean according to if the error happened or not. In my server implementation I decided to split the map rules into classes (each rule has its own class , as recommended during the lecture). I did it that way only for the 6 map rules that I have implemented , other game rules I simply check in a function that is related to those rules. I took this approach because having one method which takes map and game for example , and some classes don't care about map , or some classes don't care about the game were making my code too complicated, in a sense that I had to add gameid , and playerid to many methods so I can make a player lose a game when he does something wrong.One thing that didn't go as planned was removing expired games after 10 minutes. My problem was that if I would add the games while they were getting removed by the .remove() method I would get concurrency issues.The problem was fixed by using iterators for removal.Other thing that was that when I was trying to make difference between fortress of a player , and fortress of a enemy player , that part of the code made my halfmap combination code way more complicated.Simply because I had to add x and y coordinates of fort for each player , based on the fact how the maps are combined(my map is left , my map is right , 4x16 , 8,8).